

Homework 5: Conditional Statements and Relational Operators

CS 1323 Fall 2023

Name: _____

It is acceptable, and maybe even wise, to run these code fragments to check your work. However, I strongly encourage you to work them by hand before you run them, as this is how this type of problem will be presented on the midterm.

Student ID (usually 112-XXX-XXXX or 113-XXX-XXXX): _____

1. (12 points; 3 points each). For each part below, trace the execution of the statement by filling out the table on the right. Every change of the variable's contents should be listed, **including its initial assignment**. Tables may contain blank lines.

You will notice that I've used different styles of curly braces, to give you experience understanding the different ways that if statements can be formatted.

Although these problems can be put into eclipse to verify your answers, make sure that you work them by hand first. On the midterm exam you will not be able to use eclipse.

a)

```
int page = 97;
if (page > 50)
{
    page = 51;
}
else
{
    page = 0;
}
```

page

b)

```
int lower = 29;
int upper = 62;
if (upper < lower)
{
    int temp = lower;
    lower = upper;
    upper = temp;
}
else if (upper == lower)
{
    upper = lower + 1;
}
else
{
    upper = upper + 5;
    lower = lower - 5;
}
```

lower	upper	temp

c)

```
int size = 9;
int length = 14;
if (size == length)
{
    if (length < 5.3)
    {
        size = 8;
    }
    else
    {
        size = 6;
    }
}
else
{
    if (size > 12.2)
    {
        length = 4;
    }
    else
    {
        length = 2;
    }
}
```

length	size

d)

```
int center = 32;
int height = 20;
double left = 14;
final int BOUND = 10;
```

```
if (center >= 50) {
    if (height > center) {
        left = center + height;
    }
    else {
        left = center - height;
    }
}
else if (2*height > center) {
    if (height > center) {
        left = center - BOUND;
    }
    else {
        left = center + BOUND;
    }
}
```

left

2. (8 points; 3 points for a), 5 points for b)) The code below is supposed to implement the borderline grade provision from class. It is not correct. **Find a value for percent and finalExam that will produce an incorrect grade.** Here is the description of the borderline grade provision (from the course syllabus):

Letter Grades: Your course grade will be converted into a letter using a scale no higher than the following:

Letter	Percentage
A	90+
B	80–89
C	70–79
D	60–69
F	Below 60

Borderline Grades: It would be nice if all course grades fell cleanly into the ranges shown above. Most semesters, however, a handful of letter grades are decided by only a few points. In these difficult cases, I will use the following algorithm:

1. A course grade is considered “borderline” if it is within three points of the next higher letter. For example, 87, 68 and 79 are borderline course grades, but 81 and 92 are not.
2. For borderline grades, if the grade on the final exam is above the threshold for the higher letter, the higher letter will be given.
3. Otherwise, the lower letter will be given.

```

int percent; // This is a value you should find
int finalExam; // This is a value you should find

final int A_BORDERLINE = 90;
final int B_BORDERLINE = 80;
final int C_BORDERLINE = 70;
final int D_BORDERLINE = 60;
int tolerance = 3;

if (finalExam % 10 < 7)
{
    tolerance = 0;
}

String grade = "F"; // This is the result

if (percent + tolerance >= D_BORDERLINE)
    grade = "D";
else if (percent + tolerance >= C_BORDERLINE)
    grade = "C";
else if (percent + tolerance >= B_BORDERLINE)
    grade = "B";
else if (percent + tolerance >= A_BORDERLINE)
    grade = "A";

```

a) The values in the table below should represent an incorrect result calculated by this code.

Variable	Value
percent	
finalExam	
grade	

The best way to work on solving these problems is to read the code carefully and think about what could be wrong. Then experiment with values and see what happens.

b) Write a code fragment that calculates just the A and B grades correctly.